

Introduction

In EUROPA, planning domains and problem instances are encoded by a declarative definition language called NDDL (New Domain Definition Language). Unlike traditional definition languages such as PDDL (Planning Domain Definition Language), where a planning problem is described in the STRIPS fashion, NDDL tries to embed the concept of mutual exclusion in the language of the model. Furthermore, NDDL is an object-oriented language model.

Components

Like traditional definition language, NDDL describes a planning problem with two files: the model file, and the initial state file. The model file tries to describe dynamics of one type of problem, the common features of a benchmark problem. The initial state file, unlike its name, actually not only contains the initial state information, but also the definitions of all the variables, objects, timelines, as well as the goal state information.

Concepts

As was said in Introduction, NDDL concentrates on encoding mutual exclusion, so it has a well-defined base class called `Timeline`. Unlike other languages, the basic and most important component of NDDL is a `token`, where it has several embedded members. `start` describes the start time of the token, `end` shows the end time of the token, `duration` is the length in time of the token. Imagine the relationship of `Timeline` and `token` as follows: Suppose we have a line segment, where the `horizon` is the length of this segment. This long segment has been divided into several smaller segments, where no smaller segments overlap. These small segments are tokens. Tokens cannot be infinitely slim, i.e., every token must at least have a duration of 1.

In NDDL there is no distinction between a `state` and an `action`. Note that this is very different from classical planning concepts. In classical planning, a state represents a stable `decision point`, where various activities can happen, and the agent is left with the choice of one such activity to happen in the next time step. An action, on the other hand, represents a dynamic process, where something is going on, and the system is in the process of a state transition. If we represent a planning problem as a directed graph, then states are vertices on the graph where actions are directed edges. In NDDL every state/action is a token. People can hardly tell the difference between a state and an action if only looking at the timelines. However, states and actions are always interleaved, though not necessarily on the same timeline.

The relation of states and actions are represented by a number of `constraints`. Constraints are key components of NDDL, as they guarantee mutual exclusion, and other dependencies. An overview of constraints of NDDL can be found [here](#)